View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Universidad Distrital de la ciudad de Bogotá: Open Journal Systems

Implementation of kalman filter for the indoor location system of a lego NXT mobile robot

Leidy López Osorio*
Giovanni Bermúdez Bohórquez**
Miguel Pérez Pereira***

submitted date: March 2013
received date: April 2013
accepted date: October 2014

Abstract

This paper shows the implementation of an estimation technique based on Kalman filter to correct accumulated errors that occur along a trajectory when tracking location over a mobile platform (Lego NXT 2.0 type) in a known environment. The implementation begins with kinematic models and odometers to subsequently construct the filter and conduct the corresponding experimentation.

Key Words

Kalman Filter, kinematics, odometry, estimation, location.

1. Introduction

To apply control over a mobile platform, while it moves along a known path, it is necessary to know its behavior over time, which implies the formulation and use of models that provide such information.

The kinematic and odometer platform model provide mathematical expressions that provide the necessary information. Thus, it is possible to apply a method that uses this information in order to correct errors that may occur. This document presents an implementation of the Kalman filter to correct the ac-

* B.Sc. In Electronics; B.Sc.In Control Engineering, Universidad Distrital Francisco José de Caldas (Colombia). Current position: member of research group in Autonomous Mobile Robotics (AMRO), Universidad Distrital Francisco José de Caldas (Colombia). E-mail: lylopezo@correo.udistrital.edu.co

** B.Sc. In Electrical Engineering, Universidad Nacional de Colombia (Colombia); M.Sc. In Electronic and Computers, Universidad de Los Andes (Colombia). Current position: professor at Universidad Distrital Francisco José de Caldas. Director of research group in Autonomous Mobile Robotics (AMRO) (Colombia). E-mail: gbermudez@udistrital.edu.co

** B.Sc. In Electronic Control and Instrumentation, Universidad Distrital Francisco José de Caldas (Colombia), Specialist in Education and University Teaching, Universidad San Buenaventura (Colombia). Current position: professor at Universidad Distrital Francisco José de Caldas, member of research group in Autonomous Mobile Robotics (AMRO) (Colombia).E-mail: mrperezp@udistrital.edu.co

cumulation of positional errors during movement. This is a recursive technique that takes the kinematic model and the observations of a system and turns them into a linear model, through the Taylor expansion, in order to update the covariance of the measurements [1].

2. Implementation of Kalman Filter

The following sections provide the expressions and models that were used to implement the Kalman filter on the platform in question.

2.1. Platform

The mobile platform has been built using the LEGO MINDSTORMS NXT kit. Location information is taken from the encoders that are found in the motors and the magnetic compass, which provides a reference for orientation and is located at the top of the platform (Figure 1).

Figure 1. Mobile Platform LEGO NXT

Source: own elaboration

2.2. Kinematics and odometry

The kinematic model provides information about the platform's position and orientation at a specific time (differential platform). The parameters obtained are given by equations (1) and (2), which are defined based on physical parameters:

$$\Delta D = \frac{Xm_L - Xm_R}{2} = 0,0244(\theta_L + \theta_R) \qquad (1)$$

$$\Delta\theta = \frac{Xm_L - Xm_R}{b} = 0,2856(\theta_L - \theta_R) \qquad (2)$$

Xm_L and θ_L movement and orientation of left wheel, respectively.

Xm_R and θ_R movement and orientation of right wheel, respectively.

b distance between the two wheels.

Having a system that evolves over time (changes in position and orientation with movement) requires an expression to register this evolution and describe the location of the robot according to its own variables; see equation (3) [2].

$$X(k+1) = f(X(k), U(k)) + v(k) \qquad (3)$$

In equation (3), $X\ (k\ +1)$ represents the next position, $X\ (k)$ is the current position, $U\ (k)$ is the system input, v (k) is the vector to register non-systematic and systematic errors associated to $U\ (k)$, which is given by equation (4), where $\Delta D(k)$ is the distance the platform travels in a range $[(t]_k, t_{k+1})$, and $\Delta\theta(k)$ is the variation range orientation in the same range.

$$U(k) = [\Delta D(k)\ \Delta\theta(k)]^T \qquad (4)$$

The states of errors are given by $v(k)$, which is assumed as $v(k) \approx N(0, Q(k))$ where $Q(k)$ is the error state of the platform, which is given by equation (5).

$$Q(k) = \begin{bmatrix} k_D(D(k)\cos\theta(k)) & 0 & 0 \\ 0 & k_D(D(k)\sin\theta(k)) & 0 \\ 0 & 0 & k_{D\theta}(D(k)) + k_\theta(\theta(k)) \end{bmatrix} \quad (5)$$

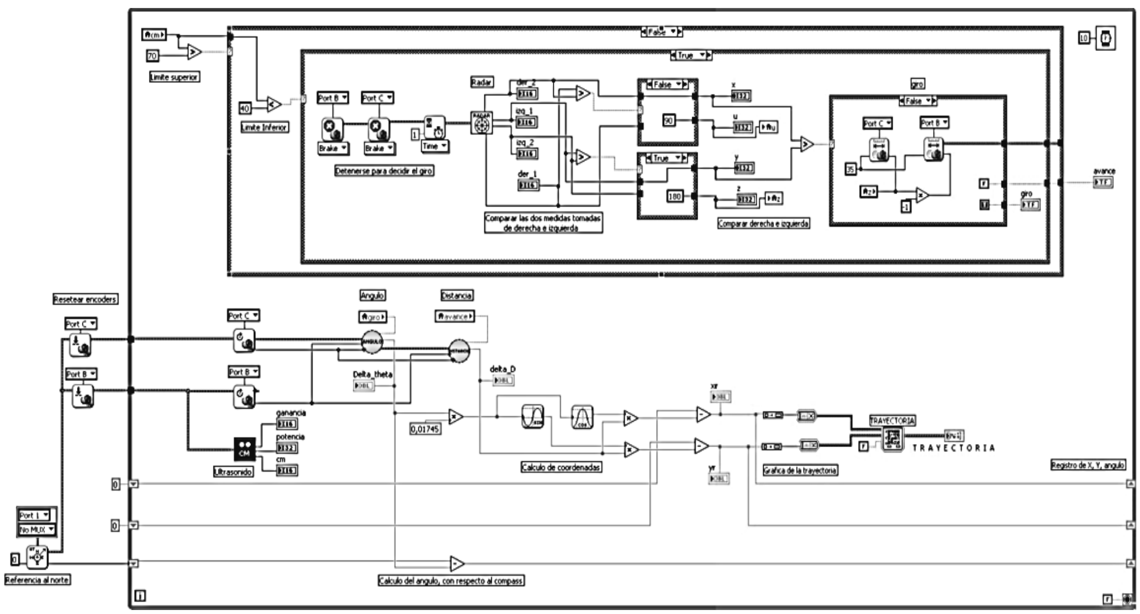
Where k_D is the coefficient error of translation of the platform related to ΔD ; $k_{D\theta}$ is the coefficient rotation error on the platform related to ΔD and k_θ is the coefficient and rotational error of the platform related to $\Delta\theta$.

Finally, the vector odometer model shown is obtained; see equation (6)

$$\begin{bmatrix} X(k+1) \\ Y(k+1) \\ \theta(k+1) \end{bmatrix} = \begin{bmatrix} x(k) + D(k)\cos\left(\theta(k) + \frac{\theta(k)}{2}\right) \\ y(k) + D(k)\sin\left(\theta(k) + \frac{\theta(k)}{2}\right) \\ \theta(k) + \frac{\theta(k)}{2} \end{bmatrix} + v(k) \quad (6)$$

After having the odometer and kinematic model, a routine (an algorithm) was formulated to record displacements and angles associated to the platform's trajectory (Figure 2).

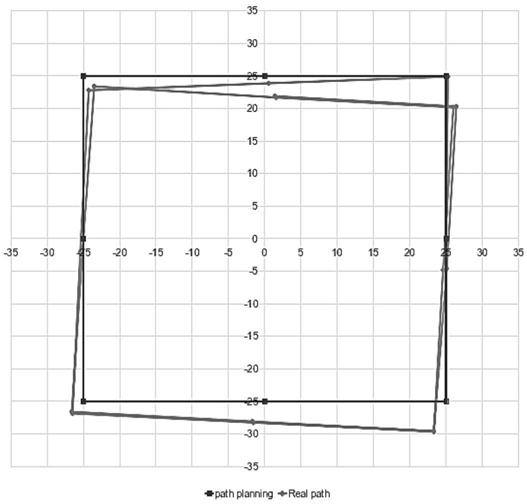
Figure 2. Algorithms for registration of distance and angle



Source: own elaboration

Figure 3 shows the results of the implementation of a route for the monitoring of a square, where the accumulation of the platform's systematic errors can be observed. [3]

Figure 3. Results of monitoring and tracking a square to calculate errors



Source: own elaboration

2.3. Kalman Filter

The Kalman filter is an estimation technique that determines the correct parameters of a system that evolves over time [4], which allows correcting errors in tracking routes.

From a series of equations, the position of the mobile platform is calculated, incorporating odometry and kinematic models. Odometry systems provide increments in the position of the center point of the relative movement to particular fixed reference axes. These increments will be taken as direct inputs to the system [5], and the kinematic model discloses the physical behavior of the mobile from its final position and orientation.

Equation (7) carries the information to locate the platform, i.e. the next estimated position.

$$X(k+1) = AX(k) + BU(k) + v(k) \tag{7}$$

Where A and B are Jacobians of the system function (equations (8) and (9)):

$$A = \begin{bmatrix} 1 & 0 & -\Delta D\sin\left(\theta + \frac{\Delta\theta}{2}\right) \\ 0 & 1 & \Delta D\cos\left(\theta + \frac{\Delta\theta}{2}\right) \\ 0 & 0 & 1 \end{bmatrix} \tag{8}$$

$$B = \begin{bmatrix} \cos\left(\theta + \frac{\Delta\theta}{2}\right) & -\Delta D\sin\left(\theta + \frac{\Delta\theta}{2}\right) \\ \sin\left(\theta + \frac{\Delta\theta}{2}\right) & \Delta D\cos\left(\theta + \frac{\Delta\theta}{2}\right) \\ 0 & 1 \end{bmatrix} \tag{9}$$

During the next estimated position for the robot, it is necessary to store the errors of the platform during movement, these errors are captured by equation (10).

$$P(k+1) = A_K P_K A_k{}^T + B_K Q_K B_k{}^T \tag{10}$$

There, $P_K = \frac{\sigma^2}{k}$, where σ is the variance of the measurements taken, k is the number of measurements taken and Q_K is the state standard error of the platform, as expressed in equation (5).

Then, to correct the position of the robot, it is necessary to supply the recorded information from the sensors (i.e. the encoders and the magnetic compass), leading to equation (11). [1]

$$z(k) = HX(k) + w(k) \tag{11}$$

Where $w(k)$ is the error associated to the measurement system and H is the Jacobian of the measurement function. Considering that this provides a direct measurement of the position and orientation of the robot, we have:

$$H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

After receiving the information provided by the sensors, it is necessary to integrate the system of estimation as follows [6]:

$$X(k+1) = X_K + K_K(Z]_K - H([X_k, 0))$$

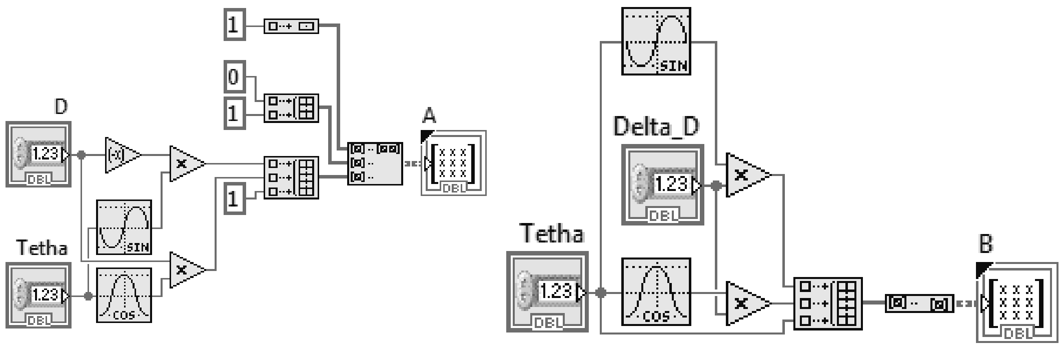
Then the error information must be updated, also including the associated measurement system, [7] [8].

$$P(k) = [(I - K]_K H_K) P_{(K+1)}$$

2.3.1. Algorithm implementation and experimentation

To implement the Kalman filter in LabVIEW, odometer and kinematic models were adapted to a stochastic state model and also to a stochastic error model. To this end, matrices for the states and the inputs of the system were constructed as shown in Figure 4.

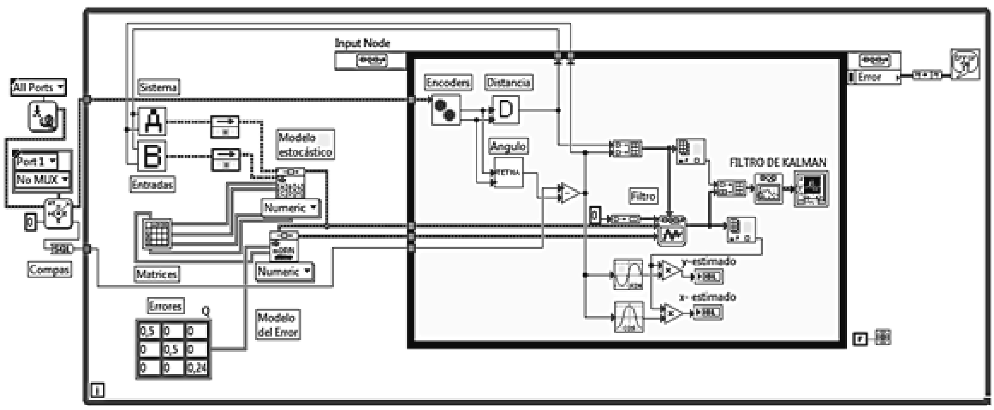
Figure 4. SubVI of the system dynamics (Jacobian A) and SubVI of the inputs system (x, y, θ) (Jacobian B)



Source: own elaboration

Once stochastic models for the state and the system error had an algorithm that would record and graph the position of the robot, using kinematics, the same action was taken to build the estimates made by the filter (Figure 5).

Figure 5. Estimation algorithm



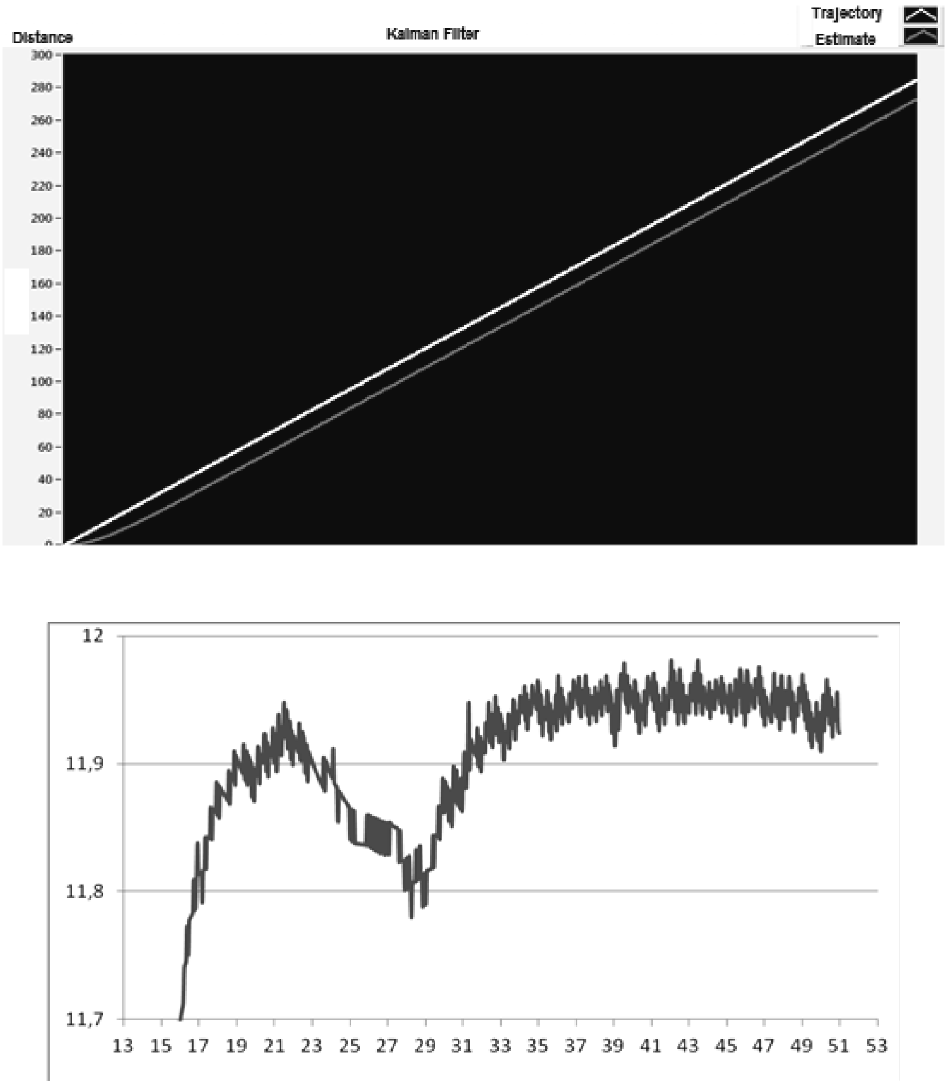
Source: own elaboration

For the experimentation process, data from three different paths were taken and the estimation error was evaluated.

- **Line**

Figure 6 shows the round distance covered by the platform in a straight line and the error between the estimate and the actual state.

Figure 6. Top: straight path; higher state of the robot using the lower estimate filter. Bottom: Error between the estimate and the state

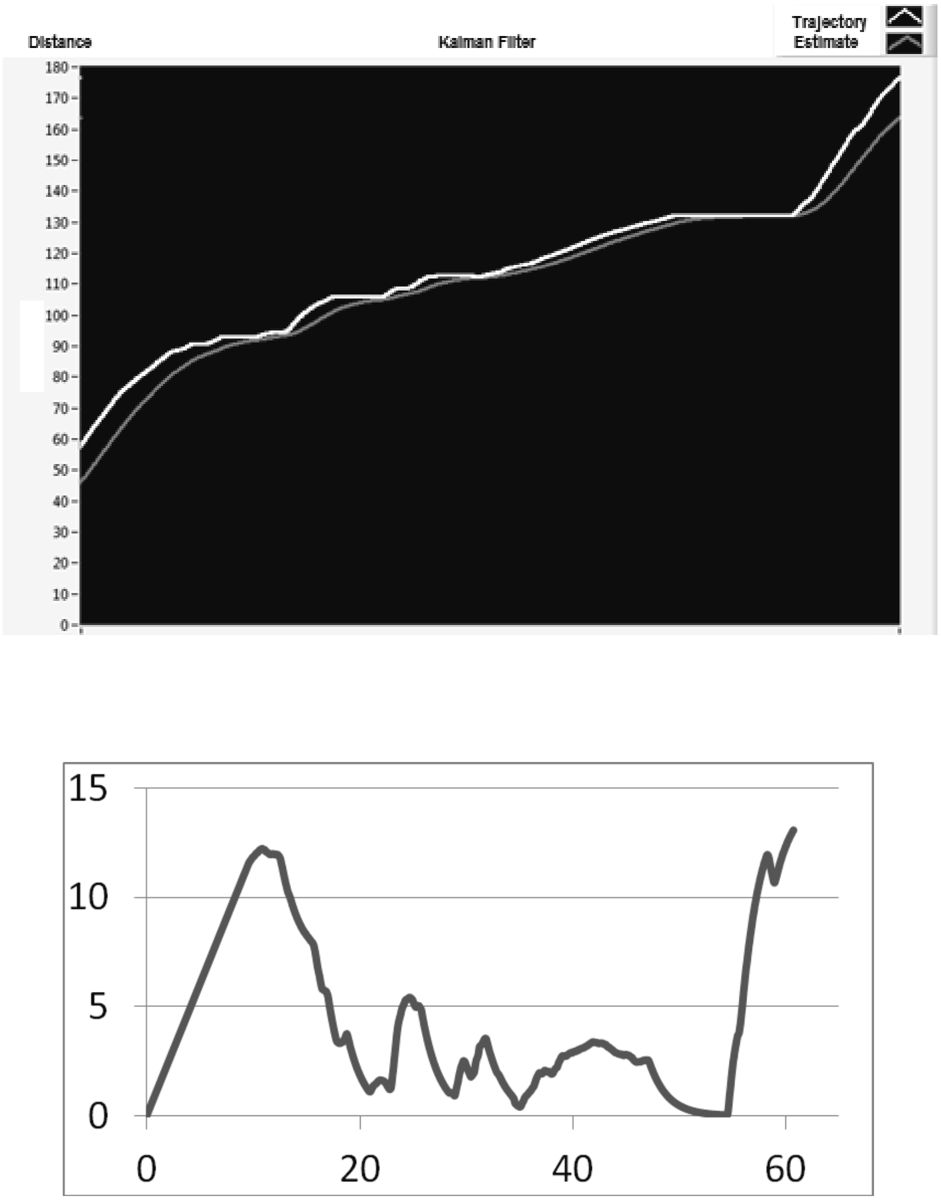


Source: own elaboration

- **Random Path**

Figure 7 shows the path followed by the robot when doing random movements caused by the entry of coordinates at random. The figure also shows the error between the estimate and the actual state for the same sample.

Figure 7. Top: higher state random robot path using the lower estimation filter. Bottom: Error between the estimate and the state

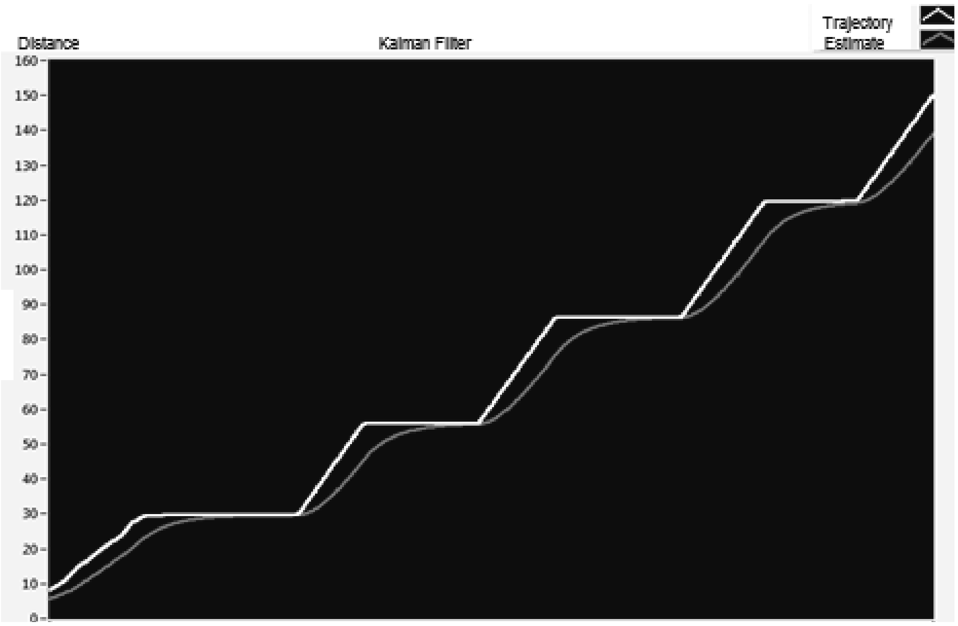


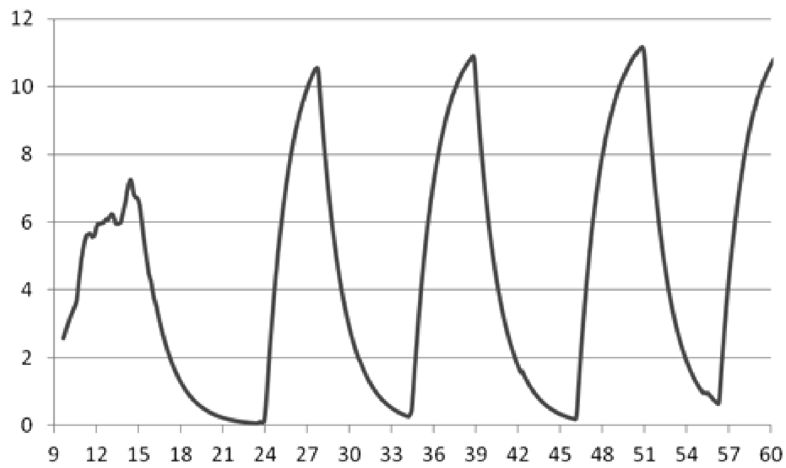
Source: own elaboration

- **Trajectory piecewise**

Figure 8 shows the round distance covered by the platform when following sections of a route. Here, one coordinate is given and then stops instantly, setting the time, and the estimation errors as well as the corresponding status.

Figure 8. Top: piecewise trajectory, upper state of the robot using the lower estimate filter. Bottom: Error between the estimate and the state





Source: own elaboration

3. Conclusions

In order to improve the performance of the mobile platform in the path of a preset route, the Kalman filter, which allowed obtaining a record of the position estimates for the robot at all times, was implemented and made it possible to have an algorithm to correct the errors in the movement, taking the values of the coordinates of the path and subtracting them from the filter results on a one-by-one basis.

The errors in the estimation of the filter can be further minimized through a similar procedure to that shown for the Lego NXT (using each analysis sensor platform with which a stochastic error approaching reality would be obtained) because in this work only the errors associated with the physical models of the robot were considered.

References

[1] M. Pinto, A. P. Moreira, and A. Matos, "Localization of Mobile Robots Using an Extended Kalman Filter in a LEGO NXT," *IEEE Transactions on Education*, pp. 1-10, 2011.

[2] G. Bermúdez, "Modelamiento cinemático y odométrico de robots móviles: aspectos matemáticos," *Tecnura*, vol. 20, no. 12, Enero 2003.

[3] P. Rodríguez M., A. Sanz M.,J.J.Pantrigo F., *"Aplicación del Filtro de Kalman al seguimiento de objetos en secuencias de imágenes",* 2012. Available in: http://www.etsii.urjc.es/asanz/documentos/MemoriaKalmanJun03.pdf, 2012

[4] F. Mart, P. Barrera, J. Mar, G. D. Rob, U. Rey y J. Carlos, "Localización basada en lógica difusa y filtros de Kalman para robots con patas", p. 12, 2006. Available in: http://gsyc.es/jmplaza/papers/cmpi2006-paco.pdf.

[5] G. Bermúdez, "Modelamiento cinemático y odométrico de robots móviles: aspectos matemáticos," *Tecnura*, vol. 20, no. 12, Enero 2003.

[6] M. I. Ribeiro and P. Lima, "Kinematics models of mobile robots," *Instituto de Sistemas e Robotica*, pp. 1000-1049, 2002.

[7] A. Ollero, *Robótica: manipuladores y robots móviles*. Marcombo, 2001.

[8] M. Pinto and G. Bermúdez, "Determinación de parámetros de un robot móvil de Lego mindstorms," *Ingenieria, Investigacion y Desarrollo*, vol. 5, no. 2, pp. 7-13, 2007.